1 Resumen de diagramas de Casos de Usos y Flujo de Datos.

1.1 Diagrama de Casos de Usos – Modulo Usuario

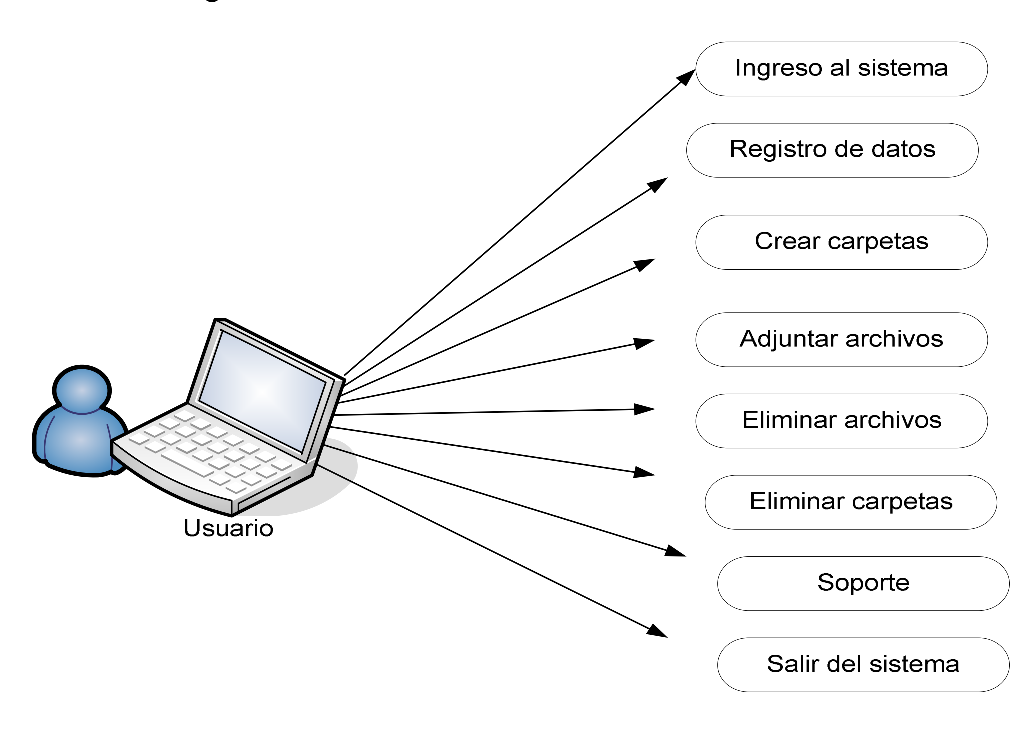


Figura # 1

1.2 Diagrama de Casos de Usos – Modulo Administrador

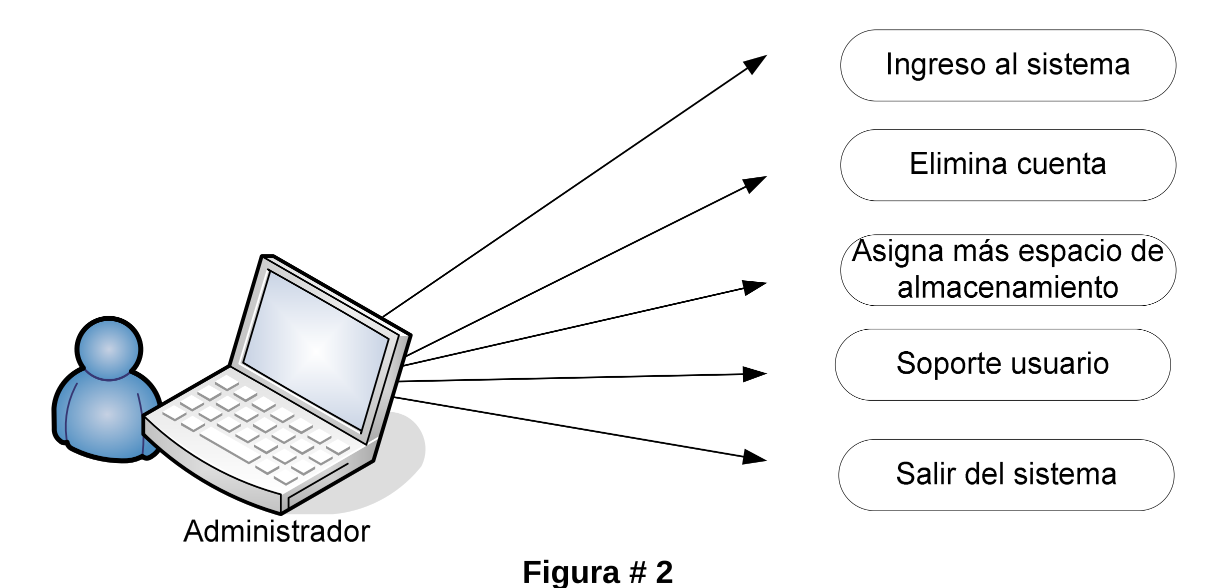


Figura # 2

2 Diagramas de Flujo de Datos y Transmisión de datos.

2.1 Diagrama de Flujo de Datos

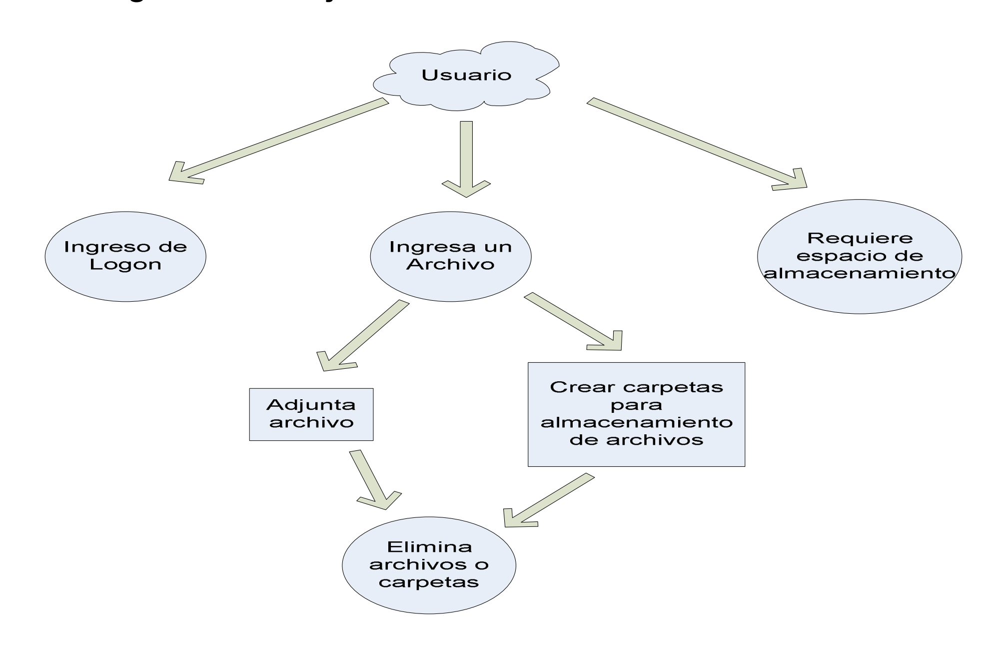


Figura # 3

2.2 Diagrama de Flujo de Transmisión de Datos

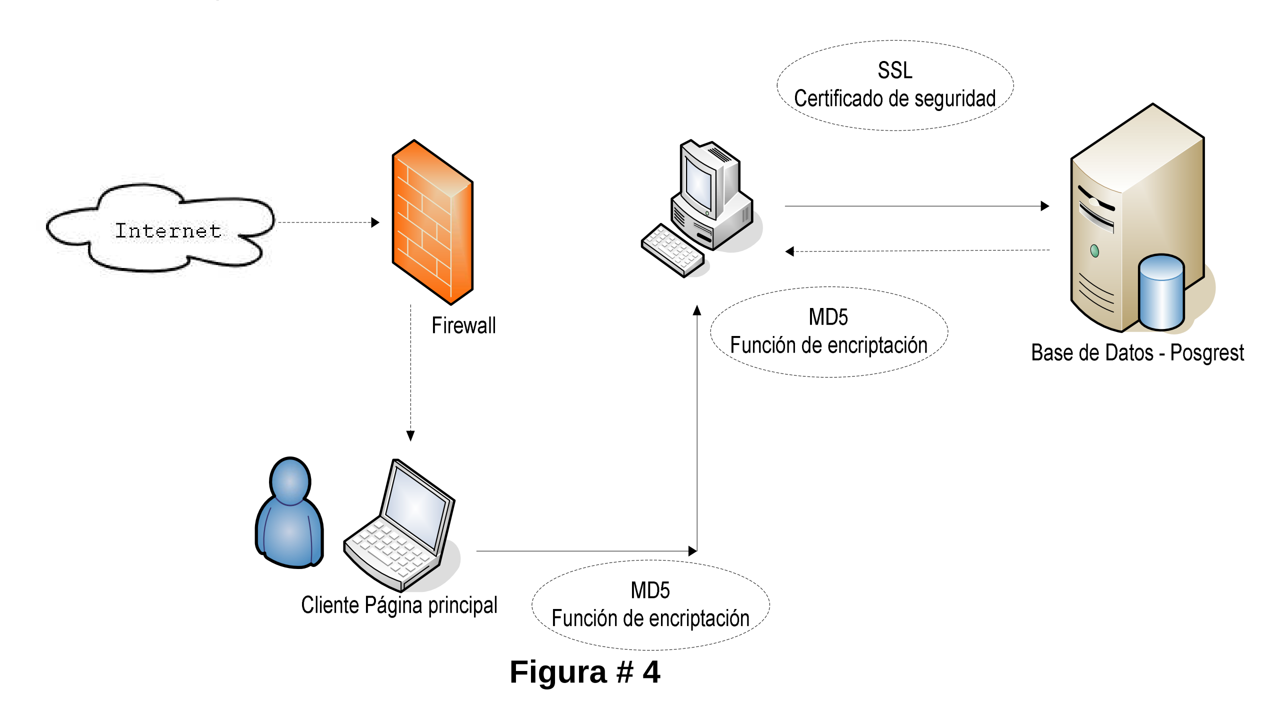


Figura # 4

3 Diagrama de Modelo Entidad – Relaciòn

3.1 Diagrama de Modelo Entidad – Relaciòn de la Aplicación.

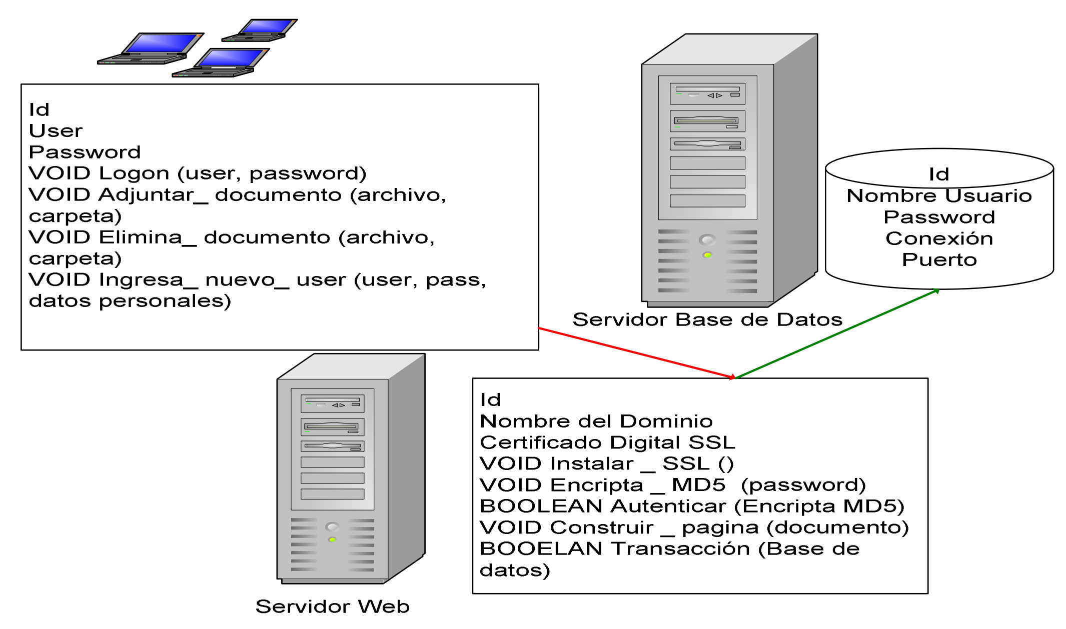
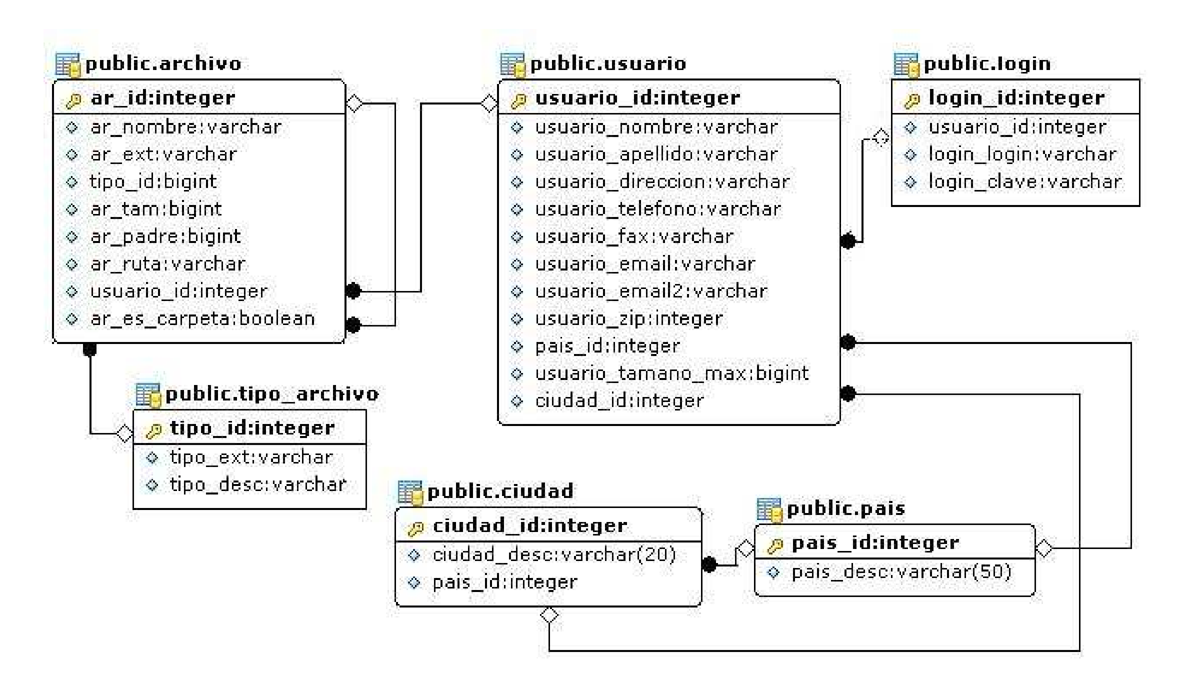


Figura # 5

3.2 Diagrama de Modelo Entidad – Relaciòn de la Base de Datos.



4 Diseño de Tablas.

4.1 Tabla Archivo

Campo	Tipo	Restricción	Descripción
ar _ id	integer	clave primaria	Número secuencial de cada uno de los archivos.
ar_ nombre	varchar	Obligatorio	Nombre del archivo.
ar_ ext	varchar	Obligatorio	Extensión del archivo.
tipo _ id	bigint	Obligatorio	Tipo del archivo.
ar_ tam	bigint	Obligatorio	Tamaño del archivo.
ar_ padre	bigint	clave foránea	Número de carpetas.
ar_ ruta	varchar	Obligatorio	Ruta del archivo.
usuario _ id	integer	clave foránea	Número de usuario.
ar_ es_ carpeta	boolean	Obligatorio	Identifica si es carpeta o archivo.

Tabla # 1

4.2 Tabla usuario

Campo	Tipo	Restricción	Descripción
usuario _ id	Integer	clave primaria	Número secuencial de cada uno de los usuarios.
usuario _ nombre	varchar	obligatorio	Nombre del usuario.
usuario _ apellido	varchar	obligatorio	Apellido del usuario.
usuario _ dirección	varchar	obligatorio	Dirección del usuario.
usuario _ teléfono	varchar	obligatorio	Número de teléfono del usuario.
usuario _ fax	varchar	obligatorio	Número de Fax del usuario.
usuario_ email	varchar	obligatorio	Dirección electrónica del usuario.
usuario _ contraseña	varchar	obligatorio	Contraseña de la dirección electrónica.
confirmación _ contraseña	varchar	obligatorio	Confirmación de contraseña del usuario.
usuario_ email2	varchar	alternativo	Dirección electrónica alternativa.
usuario_ zip	varchar	obligatorio	Còdigo de paìs
país_ id	integer	clave foránea	Número de país del usuario
usuario_ tamaño_ max	bigint	obligatorio	Asignación de espacio de memoria para cada usuario.
ciudad_ id	integer	clave foránea	Número de cuidad del usuario.

Tabla # 2

4.3 Tabla Login

Campo	Tipo	Restricción	Descripción
Login_ id	integer	clave primaria	Número secuencial de identificador de cada uno de los perfiles.
usuario_ id	integer	clave foránea	Número de usuario.
Login_ login	varchar	obligatorio	Nombre o nick del usuario.
Login_ clave	varchar	obligatorio	Clave del nombre del usuario.

Tabla # 3

4.4 Tabla Pais

Campo	Tipo	Restricción	Descripción
País_ id	Integer	clave primaria	Número secuencial del país de cada uno de los usuarios.
País_ desc	Varchar	obligatorio	Nombre del país de cada usuario.

Tabla # 4

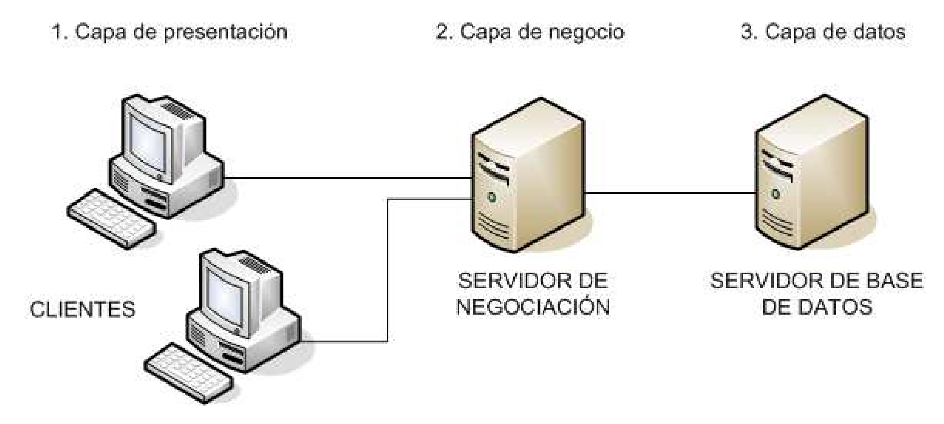
4.4 Tabla Ciudad

Campo	Tipo	Restricción	Descripción
país_ id	Integer	clave primaria	Número secuencial del país de cada uno de los usuarios.
País_ desc	Varchar	obligatorio	Nombre del país de cada usuario.

Tabla # 5

5 Arquitectura

5.1 Diseño Arquitectònico.



Figura# 7

5.2 Diagrama Arquitectònico.

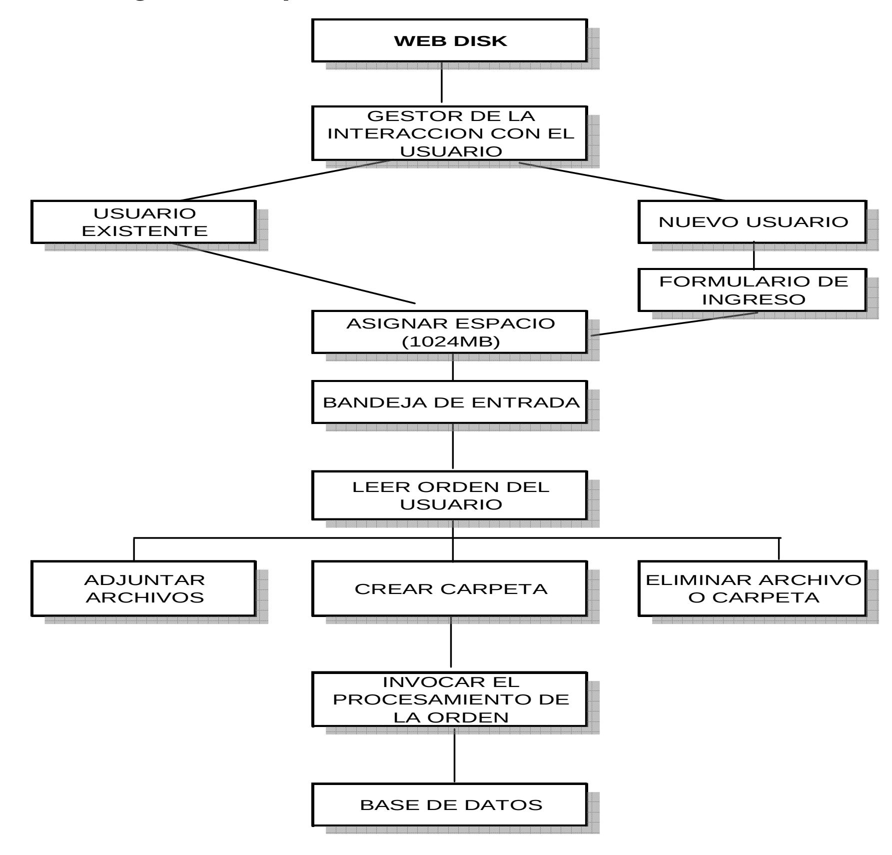


Figura # 8

6 Recursos necesarios para el Web disk

6.1 Hardware – Caracterìsticas

- ✓ Procesador Inter 3.4 ghz Hyper Thread
- ✓ 1 GB Memoria Ram - ECC Server (memoria especial para servidores)
- ✓ Mainboard Intel Soket 5775
- ✓ Disco Duro de 160 Gb serial ata
- ✓ Red inalámbrica, consiste en el de Access Point, por ejemplo el modelo108G marca DLink .
- ✓ Tarjeta de Red Giga Ethernet
- ✓ Teclado / Mouse / Parlantes
- ✓ Monitor SVGA de 15”

Nuestro servidor de tendrà su sistema operativo Linux, donde se lo configurarà con los debidos instaladores que son:

- ✓ Fedora Core 6
- ✓ SDK 6
- ✓ Sus respectivos Ipetables de comunicación
- ✓ Los RPM de Tomcat
- ✓ Protocolo TCP
- ✓ Puerto 5432

6.2 Software – Caracterìsticas.

- ✓ Paquete de Apache Tomcat 6.0.7, incluye servidor apache JDK 6.0 mas paquete Java Servlet
- ✓ Base de Datos EMS Postgrets SQL Manager 3
- ✓ Microsoft 2003 Visio
- ✓ Project
- ✓ Macromedia Dreamweaver
- ✓ Editor Java
- ✓ Plataforma de Windows XP

7 Especificaciones de Servlet utilizados.

7.1 Servlet Archivo

Este procedimiento nos permite saber si el contenido que el usuario desea guardar es un archivo o la creación de una carpeta.

```
boolean isMultipart = ServletFileUpload.isMultipartContent(req);

if (isMultipart)
if (archivoFisico(req))
devolverPaginaArchivo(resp, 1, null);            // Genera la página
else
devolverPaginaArchivo(resp, 1, error);           // Genera la página
else
if(procesarCarpeta())
devolverPaginaArchivo(resp, 2, null);            // Genera la página
else
devolverPaginaArchivo(resp, 2, error);           // Genera la página
```

Tabla # 6

Este procedimiento permitirá separar un espacio de almacenamiento de cada usuario en el que va a guardar el archivo, la subida máxima de cada archivo es de 32 Mb.

Al guardar el archivo, le asigna la ruta, si el nombre del archivo ya existe, indica que este archivo ya existía y se ha sobrescrito.

Si desea crear una carpeta se le asigna también la ruta en donde se guardará la carpeta.

```
private boolean archivoFisico(HttpServletRequest req1){
boolean a = false;
try {
DiskFileUpload fu = new DiskFileUpload();
File carpeta_temporal = new File("c://webdisk");
DiskFileItemFactory factory = new DiskFileItemFactory();
factory.setSizeThreshold(1024*1024*32); //32 Mb.
factory.setRepository(carpeta_temporal);

ServletFileUpload upload = new ServletFileUpload(factory);

upload.setSizeMax(tamano_maximo_subida);

List /* FileItem */ items = upload.parseRequest(req1);

Iterator iter = items.iterator();
while (iter.hasNext()) {
FileItem item = (FileItem) iter.next();
if (!item.isFormField()) {
String fieldName = item.getFieldName();
String fileName = item.getName();
String contentType = item.getContentType();
boolean isInMemory = item.isInMemory();
long sizeInBytes = item.getSize();
File fichero = new  File(rutaFisica() + fileName);
if (!fichero.exists()){
clArchivo = new Archivo();
clArchivo.setAr_es_carpeta(false);
clArchivo.setAr_ext(item.getName().substring(item.getName().lastIndexOf(".")+1,ite
m.getName().length()));
clArchivo.setAr_nombre(item.getName());
```

```
clArchivo.setAr_padre(Integer.parseInt(ckPadre.getValue()));
clArchivo.setAr_ruta(ckRuta.getValue());
clArchivo.setUsuario_id(Integer.parseInt(ckUsuarioCod.getValue()));
clArchivo.setAr_tam(Integer.parseInt(Long.toString(item.getSize())));
clArchivo.setTipo_id(obtenerTipo(clArchivo.getAr_ext()));
Guardar g = new Guardar(clArchivo);
a = true;
}else{
error = "El archivo ya existía y se ha sobreescrito.";
a = false;
}
item.write(fichero);
}
}

}
catch(Exception e) {
depura("Error de Aplicación " + e.getMessage());
a = false;
}
return a;
```

Tabla # 7

Este procedimiento nos permite saber el máxima ID de la tabla jornada para poder darle el siguiente ID al nuevo registro de la tabla jornada, de no existir un máximo ID por defecto se coloca 1(uno).

7.2 Servlet Datos

Este procedimiento nos permite obtener la ruta física de la carpeta principal del usuario de la webdisk.

```
private String rutaFisica(){

try{
texto = "select ar_ruta from archivo where ar_id = 1";
rs = sql.executeQuery(texto);
```

```
while (rs.next()){
webdisk = rs.getString("ar_ruta");
}
}catch(SQLException ex){
System.out.println("SQLException: "+ex.toString());
///devolverPaginaArchivo (resp);
}
String texto = webdisk+ckRuta.getValue();
//System.out.println(webdisk);
//System.out.println(ckRuta.getValue());
//System.out.println(texto);
return(texto);
}
```

Tabla # 8

Este procedimiento nos permite recuperar los cookies, donde se almacenan los últimos registros de los usuarios como: Nombre, apellido, código y logon del usuario, además la ruta física del usuario para introducirlos en la página.

```
public void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
Cookie[] arrayCookies = req.getCookies();
ckUsuario = arrayCookies[0];        // Nombre y Apellido del usuario
ckUsuarioCod = arrayCookies[1];  // Código del usuario
ckLogin = arrayCookies[2];       // Nombre de logon del usuario
ckPadre = arrayCookies[3];     // Carpeta actual(archivo) (código del archivo)
ckRuta = arrayCookies[4];      // Ruta física
ckCarpetaActual = arrayCookies[5];
click  = req.getParameter("click");
```

Tabla # 9

Este procedimiento selecciona el nombre, extensión, tipo, tamaño ruta del archivo del usuario e identifica si es carpeta con su respectiva ruta y tamaño.

Los archivos los mostrará en forma ordenada según el tipo de archivo en el que se le ha guardado.

```
try{
texto = "SELECT p.ar_ruta as ruta_padre, a.* FROM archivo a, archivo p WHERE
a.ar_id = "+ click +" and a.ar_padre = p.ar_id";
rs = sql.executeQuery(texto);
rs.next();
ar = new Archivo();
ar.setAr_id(rs.getInt("ar_id"));
ar.setAr_nombre(rs.getString("ar_nombre"));
ar.setAr_ext(rs.getString("ar_ext"));
ar.setTipo_id(rs.getInt("tipo_id"));
ar.setAr_tam(rs.getInt("ar_tam"));
ar.setAr_padre(rs.getInt("ar_padre"));
ar.setAr_ruta(rs.getString("ar_ruta"));
ar.setUsuario_id(rs.getInt("usuario_id"));
ar.setAr_es_carpeta(rs.getBoolean("ar_es_carpeta"));
rs.close();
texto = "SELECT ar_ruta FROM archivo WHERE ar_padre = " + click;
rs = sql.executeQuery(texto);
rs.next();

if (ar.getAr_es_carpeta()){
ckPadre.setValue(String.valueOf(ar.getAr_padre()));
ckRuta.setValue(ar.getAr_ruta());
rs.close();

texto = "SELECT * FROM archivo WHERE ar_padre = " + click + "ORDER BY
tipo_id";
rs = sql.executeQuery(texto);
devolverPaginaArchivo (resp);
}else{
descargarArchivo(rutaFisica()+ar.getAr_nombre(),ar.getAr_nombre(), resp);
}
rs.close();
}catch(SQLException ex){
devolverPaginaArchivo (resp);
}
}
```

Tabla # 10

7.3 Servlet Ingreso

Este procedimiento permite el ingreso del username y el password del usuario.

```
try{
texto = "SELECT   u.usuario_nombre, u.usuario_apellido, l.* FROM login l, usuario u "+
"WHERE   l.login_login = '" + usuario + "' and l.login_clave = '" + clave + "' and l.usuario_id = u.usuario_id";
rs = sql.executeQuery(texto);
ckUsuario = new Cookie("usuario","");
while(rs.next()){
ckUsuario.setValue(rs.getString("usuario_nombre")+" "+rs.getString("usuario_apellido"));
log.setLogin_id(rs.getInt("login_id"));
log.setUsuario_id(rs.getInt("usuario_id"));
log.setLogin_login(rs.getString("login_login"));
log.setLogin_clave(rs.getString("login_clave"));
}
rs.close();
```

Tabla # 11

Una vez que el usuario ingresa con su username y password, se busca sus respectivos archivos y carpetas, para mostrárselo en la bandeja de entrada.

```
if (log.getLogin_id()==0){
devolverPaginaArchivo (resp, false);
}else{
texto = "select * from archivo where ar_id = (select min(ar_id) as ar_id from archivo where ar_nombre like '%"+usuario+"%')";
rs = sql.executeQuery(texto);
ar = new Archivo();
while(rs.next()){
```

```
ar.setAr_id(rs.getInt("ar_id"));
ar.setAr_nombre(rs.getString("ar_nombre"));
ar.setAr_ext(rs.getString("ar_ext"));
ar.setTipo_id(rs.getInt("tipo_id"));
ar.setAr_tam(rs.getInt("ar_tam"));
ar.setAr_padre(rs.getInt("ar_padre"));
ar.setAr_ruta(rs.getString("ar_ruta"));
ar.setUsuario_id(rs.getInt("usuario_id"));
ar.setAr_es_carpeta(true);
}
```

Tabla # 12

7.4 Servlet Manager

Este procedimiento permitirá realizar la creación de nuevas carpetas, pero obtengo la ruta de la carpeta principal o directorio para crear otras carpetas.

```
// Pregunto si lo que recibió es un archivo o creación de carpeta
if (tipo.compareTo("archivo")==0){
boolean pudo = archivoFisico(req);
devolverPaginaArchivo(resp,1);        // Genera la página
}else
devolverPaginaArchivo(resp,2);        // Genera la página
public void procesarCarpeta(){
Archivo clArchivo = new Archivo();

clArchivo.setNombre(archivo);
clArchivo.setExtension(" ");
clArchivo.setTipo(1);
clArchivo.setTamano(0);
clArchivo.setPadre(Integer.parseInt(ckCarpeta.getValue()));
clArchivo.setRuta(ckRuta.getValue());
clArchivo.setUsuario(Integer.parseInt(ckUsuarioCod.getValue()));
Guardar g = new Guardar(clArchivo);
carpetaFisica();
//Llamada a función de creación física de la carpeta*
```

Tabla # 13

Una vez que recibimos el requerimiento de creación se procede a crearla. Al hacerlo saldrá un mensaje de si se pudo crear carpeta o no.

```
public void carpetaFisica(){
		//Obtengo la ruta de la carpeta principal

		try{
			texto = "select ar_ruta from archivo where ar_id = 1";
			rs = sql.executeQuery(texto);
			while (rs.next()){
				padre = rs.getString("ar_ruta");
			}
		}catch(SQLException ex){
			System.out.println("SQLException: "+ex.toString());
			///devolverPaginaArchivo (resp);
		}
		//mensaje = descRuta+rut.getRuta();
		mensaje = padre+ckRuta.getValue()+archivo;
		//mensaje = mensaje.substring(mensaje.lastIndexOf("
")+1,mensaje.length());
		System.out.println("Se quiere crear la carpeta"+mensaje);
		File carpeta = new File(mensaje);
		if (carpeta.mkdir()){
			System.out.println("Si se puedo crear la carpeta: "+mensaje);
		}
	}
```

Tabla # 14

7.5 Servlet Nuevo usuario

Este procedimiento permitirá ingresar datos personales a un cliente que recién desea ingresar como usuario al portal Web, permitiéndole ingresar nombre, correo, cuidad, país, etc.

```
public void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException{
		u = new Usuario();
		l = new Login();
		a = new Archivo();
```

```
		u.setUsuario_email(req.getParameter("txtEmail"));
		u.setUsuario_nombre(req.getParameter("txtNombre"));
		u.setUsuario_apellido(req.getParameter("txtApellido"));
		u.setUsuario_direccion(req.getParameter("txtDireccion"));
		u.setCiudad_id(1);
		u.setPais_id(1);
		u.setUsuario_telefono(req.getParameter("txtTelefono"));
		u.setUsuario_fax(req.getParameter("txtFax"));
		u.setUsuario_email2(req.getParameter("txtEmail2"));
		u.setUsuario_zip(req.getParameter("txtZip"));
		u.setUsuario_tamano_max(1048576);
		Guardar g = new Guardar(u);
		l.setUsuario_id(g.codigo);
		l.setLogin_login(req.getParameter("txtNick"));
		l.setLogin_clave(req.getParameter("txtPass1"));
		Guardar f = new Guardar(l);
		a.setAr_es_carpeta(true);
		a.setAr_ext("");
		a.setAr_nombre(l.getLogin_login());
		a.setAr_padre(2);
		a.setAr_tam(0);
		a.setAr_ruta("webdisk/usuarios/");
		a.setTipo_id(1);
		a.setUsuario_id(l.getUsuario_id());
		Guardar ar = new Guardar(a);
		procesarCarpeta();
		devolverPagina(resp, true);
	}

	private void procesarCarpeta(){
		try{
			texto = "select ar_ruta from archivo where ar_id = 1";
			rs = sql.executeQuery(texto);
			while (rs.next()){
				texto = rs.getString("ar_ruta");
			}
		}catch(SQLException ex){
			System.out.println("SQLException: "+ex.toString());
		}
		texto = texto+a.getAr_ruta()+a.getAr_nombre();
		File carpeta = new File(texto);
		if (!carpeta.exists()){
			if (carpeta.mkdir()){}
		}else{
			System.out.println("Ya existe la carpeta, no se puede crear.");
```

Tabla # 15

7.6 Servlet Usuario

Este procedimiento devuelve la página de la bandeja de entrada de cada usuario.

```
private long getFileSize(File folder){
long foldersize = 0;
File[] filelist = folder.listFiles();
for (int i = 0; i < filelist.length; i++){
if (filelist[i].isDirectory()){
foldersize += getFileSize(filelist[i]);
}else{
foldersize += filelist[i].length();
} }
return foldersize;
}
private String tamanoCarpeta(){
try{
texto = "select ar_ruta from archivo where ar_id = 1";
rs = sql.executeQuery(texto);
while (rs.next()){
webdisk = rs.getString("ar_ruta");
}
}catch(SQLException ex){
System.out.println("SQLException: "+ex.toString());
}
ruta = webdisk+"webdisk/usuarios/"+ckLogin.getValue()+"/";
System.out.println(ruta);
File archivo = new File(ruta);
long t = getFileSize(archivo);
return convertirBytes(String.valueOf(t));
}
private void devolverPagina (HttpServletResponse resp){
resp.addCookie(ckUsuario);
resp.addCookie(ckUsuarioCod);
resp.addCookie(ckLogin);
//ckPadre.setValue(click);
resp.addCookie(ckPadre);
//ckRuta.setValue(ar.getAr_ruta()+ar.getAr_nombre()+"/");
resp.addCookie(ckRuta);
resp.addCookie(ckCarpetaActual);
String nombre;
PrintWriter out = null;
try{
out = resp.getWriter();
}catch (IOException io){
System.out.print("Error al escribir la pagina"+io.getMessage()); }
```

Tabla # 16

8 Beans.

Son los objetos que interactúan con los servelt, a continuación detallamos los siguientes:

8.1 Usuario

```
public void setUsuario_id(int usuario_id) { this.usuario_id = usuario_id; }
public void setUsuario_nombre(String usuario_nombre) { this.usuario_nombre =
usuario_nombre; }
public void setUsuario_apellido(String usuario_apellido) { this.usuario_apellido =
usuario_apellido; }
public void setUsuario_direccion(String usuario_direccion) { this.usuario_direccion =
usuario_direccion; }
public void setUsuario_telefono(String usuario_telefono) { this.usuario_telefono =
usuario_telefono; }
public void setUsuario_fax(String usuario_fax) { this.usuario_fax = usuario_fax; }
public void setUsuario_email(String usuario_email) { this.usuario_email =
usuario_email; }
public void setUsuario_email2(String usuario_email2) { this.usuario_email2 =
usuario_email2; }
public void setUsuario_zip(String usuario_zip) { this.usuario_zip = usuario_zip; }
public void setPais_id(int pais_id) { this.pais_id = pais_id; }
public void setUsuario_tamano_max(int usuario_tamano_max) {
this.usuario_tamano_max = usuario_tamano_max; }
public void setCiudad_id(int ciudad_id) { this.ciudad_id = ciudad_id; }
public int getUsuario_id() { return (this.usuario_id); }
public String getUsuario_nombre() { return (this.usuario_nombre); }
public String getUsuario_apellido() { return (this.usuario_apellido); }
public String getUsuario_direccion() { return (this.usuario_direccion); }
public String getUsuario_telefono() { return (this.usuario_telefono); }
public String getUsuario_fax() { return (this.usuario_fax); }
public String getUsuario_email() { return (this.usuario_email); }
public String getUsuario_email2() { return (this.usuario_email2); }
public String getUsuario_zip() { return (this.usuario_zip); }
public int getPais_id() { return (this.pais_id); }
public int getUsuario_tamano_max() { return (this.usuario_tamano_max); }
public int getCiudad_id() { return (this.ciudad_id); }
public String getCampos()
```

Tabla # 17

8.2 Archivo

```
public Archivo(){
this.ar_id = 0;
this.ar_nombre = "";
this.ar_ext = "";
this.tipo_id = 0;
this.ar_tam = 0;
this.ar_ruta = "";
this.ar_padre = 0;
this.usuario_id = 0;
this.ar_es_carpeta = false;
}
public Archivo(int ar_id, String ar_nombre, String ar_ext,
int tipo_id, int ar_tam, int ar_padre, String ar_ruta, int usuario_id,
boolean ar_es_carpeta) {
this.ar_id = ar_id;
this.ar_nombre = ar_nombre;
this.ar_ext = ar_ext;
this.tipo_id = tipo_id;
this.ar_tam = ar_tam;
this.ar_padre = ar_padre;
this.ar_ruta = ar_ruta;
this.usuario_id = usuario_id;
this.ar_es_carpeta = ar_es_carpeta;
}
public void setAr_id(int ar_id) { this.ar_id = ar_id; }
public void setAr_nombre(String ar_nombre) { this.ar_nombre = ar_nombre; }
public void setAr_ext(String ar_ext) { this.ar_ext = ar_ext; }
public void setTipo_id(int tipo_id) { this.tipo_id = tipo_id; }
public void setAr_tam(int ar_tam) { this.ar_tam = ar_tam; }
public void setAr_padre(int ar_padre) { this.ar_padre = ar_padre; }
public void setAr_ruta(String ar_ruta) { this.ar_ruta = ar_ruta; }
public void setUsuario_id(int usuario_id) { this.usuario_id = usuario_id; }
public void setAr_es_carpeta(boolean ar_es_carpeta) { this.ar_es_carpeta =
ar_es_carpeta; }
public int getAr_id() { return (this.ar_id); }
public String getAr_nombre() { return (this.ar_nombre); }
public String getAr_ext() { return (this.ar_ext); }
public int getTipo_id() { return (this.tipo_id); }
public int getAr_tam() { return (this.ar_tam); }
public int getAr_padre() { return (this.ar_padre); }
public String getAr_ruta() { return (this.ar_ruta); }
public int getUsuario_id() { return (this.usuario_id); }
public boolean getAr_es_carpeta() { return (this.ar_es_carpeta); }
```

Tabla # 18

8.3 Ciudad

```
public class Ciudad {
private int ciudad_id = 0;
private String ciudad_desc = "";
private int pais_id = 0;

public Ciudad(int ciudad_id, String ciudad_desc, int pais_id) {
this.ciudad_id = ciudad_id;
this.ciudad_desc = ciudad_desc;
this.pais_id = pais_id;
}
public Ciudad(Ciudad other) {
if(this != other) {
this.ciudad_id = other.ciudad_id;
this.ciudad_desc = other.ciudad_desc;
this.pais_id = other.pais_id;
}
}
public void setCiudad_id(int ciudad_id) { this.ciudad_id = ciudad_id; }
public void setCiudad_desc(String ciudad_desc) { this.ciudad_desc =
ciudad_desc; }
public void setPais_id(int pais_id) { this.pais_id = pais_id; }
public int getCiudad_id() { return (this.ciudad_id); }
public String getCiudad_desc() { return (this.ciudad_desc); }
public int getPais_id() { return (this.pais_id); }
public String toString() {
String sep = System.getProperty("line.separator");
StringBuffer buffer = new StringBuffer();
buffer.append(sep);
buffer.append("ciudad_id = ");
buffer.append(ciudad_id);
buffer.append(sep);
buffer.append("ciudad_desc = ");
buffer.append(ciudad_desc);
buffer.append(sep);
buffer.append("pais_id = ");
buffer.append(pais_id);
buffer.append(sep);

return buffer.toString();
}
}
```

Tabla # 19

8.4 Guardar

```
public Guardar(Usuario usu){
this.Conectar();
try{
texto = "INSERT INTO usuario ("+usu.getCampos()+")
VALUES("+usu.getValores()+")";
Sql.executeUpdate(texto);
texto = "select max(usuario_id) as usuario_id from usuario";
rs = Sql.executeQuery(texto);
System.out.println("Si paso");
while(rs.next()){
this.codigo = rs.getInt("usuario_id");
}
System.out.println("----"+this.codigo+"-----");
}
catch(SQLException ex){
System.out.println("SQLException: "+ex.getMessage());
}  }
public Guardar(Login log){
this.Conectar();
try{
texto = "INSERT INTO login ("+log.getCampos()+")
VALUES("+log.getValores()+")";
Sql.executeUpdate(texto);
texto = "select max(login_id) as login_id from login";
//System.out.println(texto);
rs = Sql.executeQuery(texto);
while(rs.next()){
this.codigo = rs.getInt("login_id");
}  }
catch(SQLException ex){
System.out.println("SQLException: "+ex.getMessage());
}  }
public Guardar(Tipo tip){
this.Conectar();
try{
texto = "INSERT INTO tipo_archivo ("+tip.getCampos()+")
VALUES("+tip.getValores()+")";
Sql.executeUpdate(texto);
texto = "select max(tipo_id) as tipo_id from tipo_archivo";
rs = Sql.executeQuery(texto);
while(rs.next()){
this.codigo = rs.getInt("tipo_id");
}   }
```

Tabla # 20

8.5 Login

```
public class Login{
private int login_id;
private int usuario_id;
private String login_login;
private String login_clave;

public Login(){
this.login_id = 0;
this.usuario_id = 0;
this.login_login = "";
this.login_clave = "";
}

public Login(int login_id, int usuario_id, String login_login,
String login_clave) {
this.login_id = login_id;
this.usuario_id = usuario_id;
this.login_login = login_login;
this.login_clave = login_clave;
}
public Login(Login other) {
if(this != other) {
this.login_id = other.login_id;
this.usuario_id = other.usuario_id;
this.login_login = other.login_login;
this.login_clave = other.login_clave;
}
}
public void setLogin_id(int login_id) { this.login_id = login_id; }
public void setUsuario_id(int usuario_id) { this.usuario_id = usuario_id; }
public void setLogin_login(String login_login) { this.login_login = login_login; }
public void setLogin_clave(String login_clave) { this.login_clave = login_clave;
}

public int getLogin_id() { return (this.login_id); }
public int getUsuario_id() { return (this.usuario_id); }
public String getLogin_login() { return (this.login_login); }
public String getLogin_clave() { return (this.login_clave); }
```

Tabla # 21

8.6 Obtener

```
public Archivo getArchivo(int id){
Archivo arc = null;
try{
texto = "SELECT * FROM archivo WHERE ar_id = "+id;
rs = sql.executeQuery(texto);
while(rs.next()){
arc.setAr_id(rs.getInt("ar_id"));
arc.setAr_nombre(rs.getString("ar_nombre"));
arc.setAr_ext(rs.getString("ar_ext"));
arc.setTipo_id(rs.getInt("tipo_id"));
arc.setAr_tam(rs.getInt("ar_tam"));
arc.setAr_ruta(rs.getString("ar_ruta"));
arc.setAr_padre(rs.getInt("ar_padre"));
arc.setAr_es_carpeta(rs.getBoolean("ar_es_carpeta"));
arc.setUsuario_id(rs.getInt("usuario_id"));
}
}catch (SQLException ex){
System.out.println("SQLException: "+ex.getMessage());
}
return arc;
}
public Ruta getRuta(int id){
Ruta rut = new Ruta();
return rut;
}
public Tipo getTipo(int id){
Tipo tip = new Tipo();
return tip;
}
public Tipo getTipo(String ext){
Tipo tip = new Tipo();
int codigo=0;
try{   con = DriverManager.getConnection(url,"","");
sql = con.createStatement();
texto = "SELECT * FROM tipo_archivo WHERE tipo_ext LIKE UPPER('%" + ext + "%')";
//System.out.println(texto);
rs = sql.executeQuery(texto);
while(rs.next()){  tip.setTipo_id(rs.getInt("tipo_id"));
tip.setTipo_desc(rs.getString("tipo_desc"));
tip.setTipo_ext(rs.getString("tipo_ext"));
}  if (tip.getTipo_id()==0){
System.out.println(texto);
//tip.setTipo_desc("");
tip.setTipo_ext(ext.toUpperCase());
Guardar g = new Guardar(tip);
tip.setTipo_id(g.getCodigo());  }
```

Tabla # 22

8.7Paìs

```
public class Pais {
	private int pais_id = 0;
	private String pais_desc = "";


	public Pais(int pais_id, String pais_desc) {
		this.pais_id = pais_id;
		this.pais_desc = pais_desc;
	}

	public Pais(Pais other) {
		if(this != other) {
			this.pais_id = other.pais_id;
			this.pais_desc = other.pais_desc;
		}
	}

	public void setPais_id(int pais_id) { this.pais_id = pais_id; }
	public void setPais_desc(String pais_desc) { this.pais_desc =
pais_desc; }

	public int getPais_id() { return (this.pais_id); }
	public String getPais_desc() { return (this.pais_desc); }

	public String toString() {

		String sep = System.getProperty("line.separator");

		StringBuffer buffer = new StringBuffer();
		buffer.append(sep);
		buffer.append("pais_id = ");
		buffer.append(pais_id);
		buffer.append(sep);
		buffer.append("pais_desc = ");
		buffer.append(pais_desc);
		buffer.append(sep);

		return buffer.toString();
	}
}
```

Tabla # 23

9. DICCIONARIO DE DATOS

9.1 Tipo de datos enteros en Java:

Tipo	Tamaño (bytes)	Rango
int	4	-2147483648 a 2147483647
short	2	-32768 a 32767
long	8	-9223372036854775808 a 9223372036854775807
byte	1	-128 a 127

Tabla # 24

9.2 Tipo de datos reales en Java:

Tipo	Tamaño (bytes)	Rango
float	4	+ 3.40282347E+38
Double	8	+ 179769313486231570E+308

Tabla # 25

9.3 Operadores lógicos:

Operador	Operación que realiza
!	Not lógico
==	Test de igualdad
!=	Test de desigualdad
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
&&	And lógico
||	Or lógico

Tabla # 26

9.4 Términos empleados:

Término	Significado
Primary key	Indica la clave primaria
Foreign key	Indica la clave foránea
Not null	Indica que un campo es obligatorio
Public	Declara pública una clase, variable o método
Class	Para declarar una clase
Private	Declara privada una clase, variable o método
String	Tipo de dato (cadena de caracteres)
Null	Que no tiene valor
Boolean	Tipo de dato (solo tiene 2 valores: true y false)
language="java"	Comunica al servidor el lenguaje que va a ser utilizado en el archivo.
extends="package.class"	La variable extends, define la clase padre del servlet generado.
import="package.*,package.class"	Sirve para especificar los paquetes y clases que se quieran utilizar.
Request	Es la petición del cliente. Es normalmente una subclase de la case HttpServletRequest.
Response	Es la página JSP de respuesta y es una subclase de HttpServletResponse.
Exception	Es una subclase libre de Throwable que es pasada a la página que majena los errores.
Commit	Cuando quiero que las instrucciones se ejecuten.

Roll back	Cuando no quiero que se ejecuten las instrucciones.
Create	Utilizado para crear nuevas tablas, campos e índices.
Select	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado
Insert	Utilizado para cargar lotes de datos en la base de datos en una única operación.
Update	Utilizado para modificar los valores de los campos y registros especificados.
Delete	Utilizado para eliminar registros de una tabla de una base de datos.
From	Utilizada para especificar la tabla de la cual se van a seleccionar los registros.
Implements	Clase que contiene la implementación de todos los métodos de una o más interfaces.
java.lang	clases esenciales, números, strings, objetos, compilador, runtime, seguridad y threads
java.io	clases que manejan entradas y salidas
java.util	clases útiles, como estructuras genéricas, manejo de fecha, hora y strings, número aleatorios, etc.

Tabla # 27

1. ANTECEDENTES DEL MANUAL DE USUARIO

La Base de Datos de archivos encriptados vía Web o "**Web Disk**", es un sistema enfocado al almacenamiento de archivos online que puede guardar cualquier tipo de información. Los usuarios pueden distribuir y respaldar archivos individuales, carpetas o directorios enteros.

A continuación se detallarán cada una de las pantallas; así también como el funcionamiento de las mismas con el fin de lograr mayor captación del usuario y por supuesto mejor comprensión en el manejo de nuestro sistema.

2. BIENVENIDA

El Portal da una bienvenida al usuario, indicándole los beneficios que obtendrá al lograr suscribirse en nuestra Web, además que es gratuita, tiene la opción de tener un espacio en nuestro disco virtual de un 1Gb para guardar toda la información importante y confidencial, como observamos en la figura # 1.

El Menú principal del sistema es presentado a cualquier usuario que desee conocer acerca del Portal, de igual manera conocerá los requisitos y

requerimientos que se solicitan para formar parte de nuestros clientes y mediante las preguntas frecuentes despejará cualquier duda acerca del Portal.

Esta es la bienvenida que da nuestra Web a todos los usuarios que desean tener un disco virtual gratuito.

Bienvenido a webDisk

Es una herramienta que te permite almacenar archivos en la red y consultarlos desde cualquier lugar. Es similar a un disco rigido, con la ventaja que puedes usarlo desde cualquier computadora, simplemente conectándote a Internet.
Con el webDisk podrás almacenar archivos, accederlos mediante una dirección web desde cualquier lugar y compartirlos, a través de usuario y contraseña.

Novedades webDisk

Además de disponer de 1 GigaByte para su uso por cada usuario, puedes revisar tu informacion de la misma manera que en el explorador de windows para mayor comodidad y facil distribucion de los archivos y carpetas. mas info..

Fig. 1 Pagina Bienvenida

3. Pagina de sesión de inicio de usuarios

ACERCA DE
PUBLICIDAD
SOPORTE
CONTACTENOS
AYUDA ?
webDisk
Servicio de Almacenamiento Virtual
UNIVERSIDAD DE GUAYAQUIL
Bienvenido a webDisk
Es una herramienta que te permite almacenar archivos en la red y consultarlos desde cualquier lugar. Es similar a un disco rígido, con la ventaja que puedes usarlo desde cualquier computadora, simplemente conectándote a Internet.
Con el webDisk podrás almacenar archivos, accederlos mediante una dirección web desde cualquier lugar y compartirlos, a través de usuario y contraseña.
Novedades webDisk
Además de disponer de 1 GigaByte para su uso por cada usuario, puedes revisar tu informacion de la misma manera que en el explorador de windows para mayor comodidad y facil distribucion de los archivos y carpetas. mas info..
Acceso de usuario
Proporcione información de inicio de sesión.
Nombre de usuario *
Contraseña *
Recordar mi nombre de usuario
¿Ha olvidado su contraseña?
Iniciar Sesión
Registro
El registro y envío de su información personal facilitará su uso de nuestro sitio.
Registrar

Fig. 2 Pagina de inicio

Si todavía no es cliente regístrese a nuestra Webdisk, dando un clic en registrar, luego aparecerá la figura # 3.

Si el visitante tiene dudas e inquietudes sobre ¿Qué es el Webdisk?, Cómo puede acceder?, Si tiene algún costo por el servicio?, Cómo funciona?, entre

otras. Tiene en el menú la opción de ayuda, que le permite solventar todas sus dudas; pues, dentro de Preguntas Frecuentes encontrará las respuestas a ellas.

4. Formulario de Ingreso de Usuario

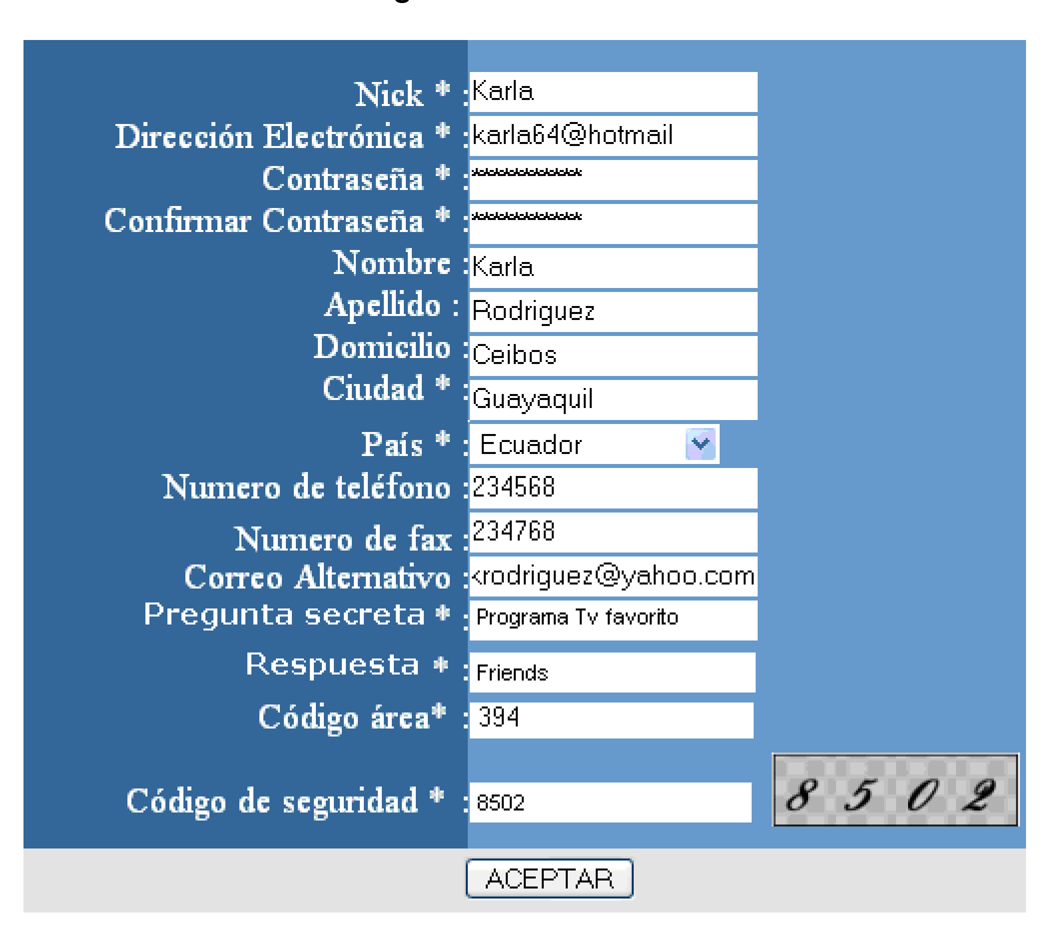


Figura # 3

En este formulario se registra todos los datos del usuario, como puede observar tiene que llenar cada caja de texto la información, esto ayudara a lo posterior al Administrador mantenerse en contacto con usted y poder resolver cualquier inquietud que usted tenga.

4.1 Datos del formulario

Esta ventana encontraremos el siguiente cuadro de texto donde ingresaremos los datos personales del usuario

Nick: Se ingresará el Login del usuario en el sistema.

Dirección Electrónica: Ingresara el correo electrónico que posee el usuario.

Contraseña: Se ingresara el password

Confirmar contraseña: Ingresara nuevamente el password ingresado anteriormente.

Nombre: Se ingresará el nombre del usuario

Apellido: Se ingresará el apellido del usuario

Domicilio: Ingresara la dirección de domicilio u oficina.

País: Aparece una lista despegable del país donde radica el usuario

Ciudad: Escogerá de la lista despegable la ciudad donde se encuentra el usuario.

Numero de fax: Indicara el número de fax o teléfono del usuario.

Código de Área: ingresa el código de área del país.

Correo Alternativo: Ingresara otro correo electrónico que posee el usuario

Pregunta secreta: Digite una pregunta secreta, esto ayudara en caso si se olvida la contraseña.

Respuesta: Digite la respuesta de su pregunta secreta, esto ayudara a identificar que usted es el usuario.

Código de seguridad: ingrese el número que se genera en la imagen, esto es una seguridad para usted, ya que es un único número que se genera para cada usuario. Nadie excepto el administrador tendrá sus datos confidenciales.

Luego de dar clic en aceptar. Retornara automáticamente a la figura #2 (Pagina de inicio) donde ingresara en la caja de texto el nombre de usuario y contraseña, el usuario procederá a ingresar el logon y la clave respectivamente que le servirá para iniciar su sesión dentro del sistema.

En caso que el usuario no teclee ningún dato aparecerá una alerta que indica "Por favor, introduzca un nombre de usuario".



Figura # 4

5. Recuperación de Contraseña

Si el usuario no recordare su contraseña, ingresara la opción de la recuperación de contraseña, ingresara en el cuadro de texto su Nombre y correo electrónico para que sea enviado una nueva contraseña y poder ingresar nuevamente a sus datos, como indica la figura # 5

Recuperación de contraseña

¿Ha olvidado su contraseña?

Introduzca su nombre de usuario para recibir su contraseña.

Correo electronico

Pregunta Secreta

Respuesta

Ingrese Nueva contraseña

Enviar

Ir a la página de inicio de sesión?

Figura # 5

6. Bandeja de Entrada

Una vez registrado el usuario en el Web Disk, podrá ingresar a la bandeja de Entrada, donde guardara sus archivos, las podrá mover a cualquier carpeta y también crear carpetas como usted lo desee.

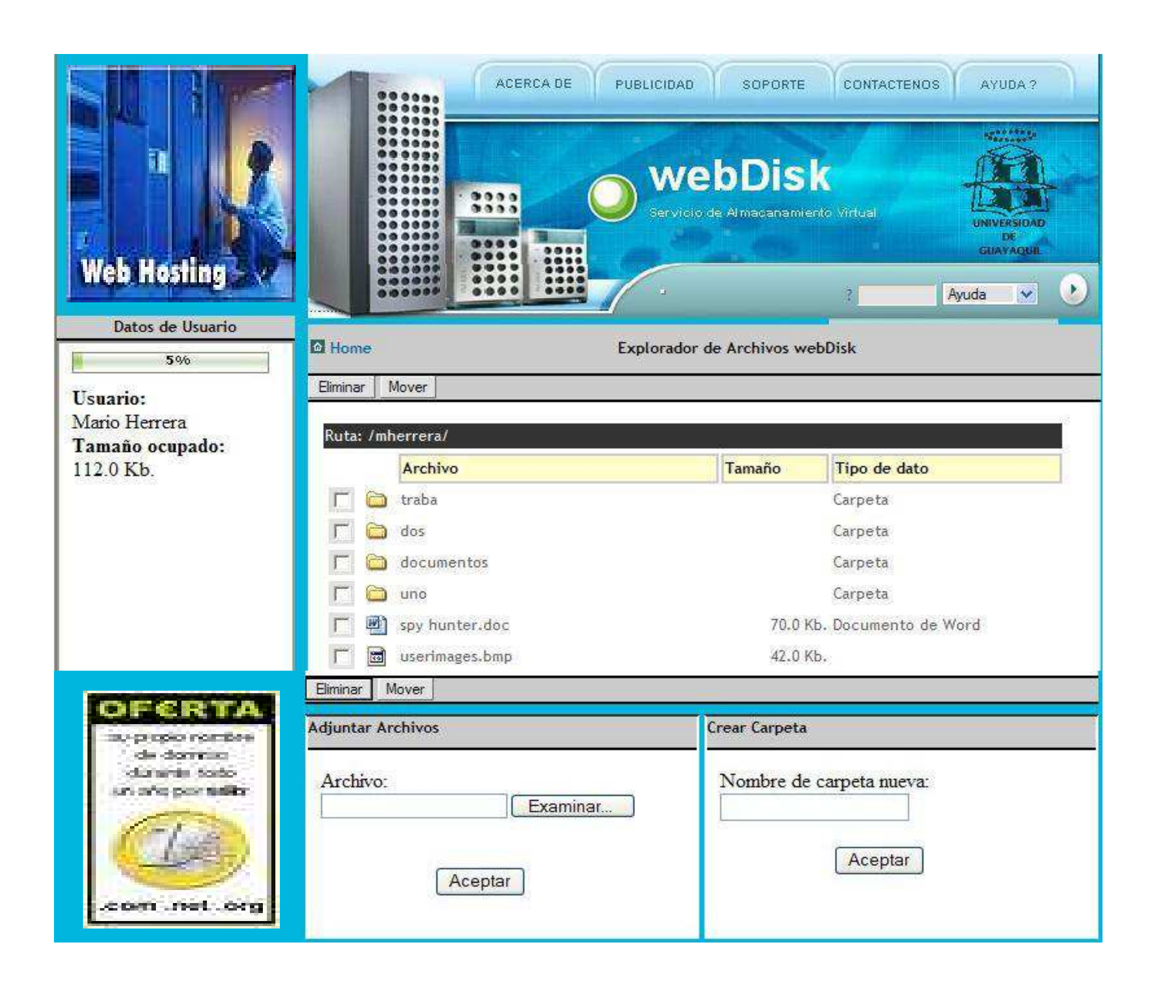


Figura # 6

Al costado derecho estará el nombre del usuario, los menús correspondientes del usuario, el mismo que podrá hacer clic en cualquiera de ellos para elaborar el trabajo que requiera en el sistema.

7. Adjuntar Archivo

Permite al usuario adjuntar archivos, haga un clic seleccionando el archivo que desea almacenar en la webdisk desde cualquier dispositivo de almacenamiento, ya sea en un pendrive, CDS, o en el disco duro de la computadora que esta realizando la operación. Asegúrese de tener espacio suficiente en el disco virtual para almacenar los archivos.

Una vez aceptado aparecerá el archivo adjuntado por el usuario, luego surgirá el archivo adjunto como observamos en la figura # 7.

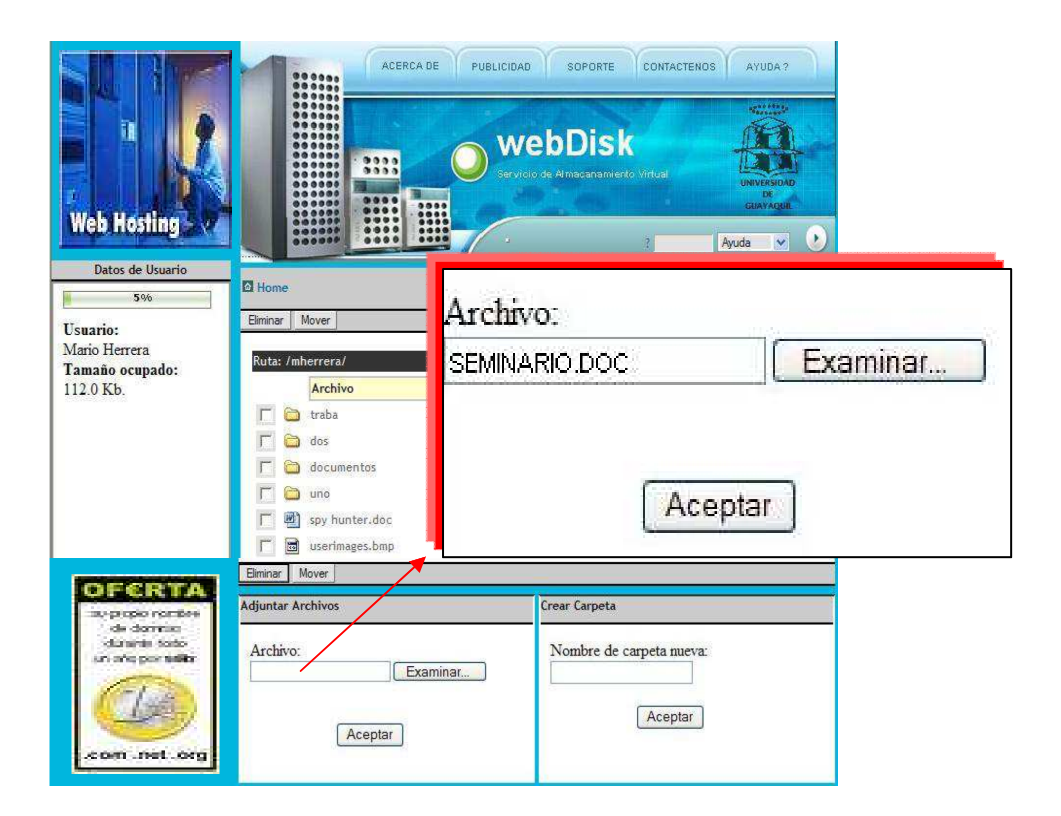


Figura # 7

8. Creación de Carpetas

El usuario podrá crear las carpetas que desee, y adjuntar los archivos en las carpetas correspondientes, así tendrá organizada y en orden sus documentos.

Para crear una carpeta, escribo en el cuadro de texto el nombre de la carpeta y doy clic en aceptar, luego aparecerá en mi bandeja de entrada la carpeta creada, como se demuestra en la figura # 8

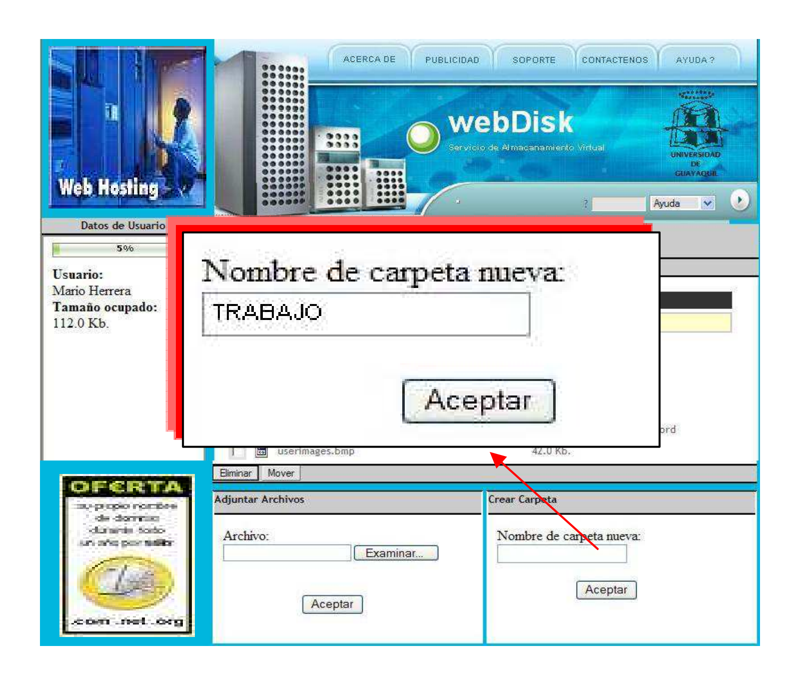


Figura # 8

9. Menu de la Webdisk

Por medio de este menú, situado en la parte posterior izquierda de la página, los usuarios podrán ver su espacio de almacenamiento, solicitar mas espacio de memoria, consultar soporte y cerrar la sesión, como demuestra la figura # 9. La barra de almacenamiento indicara el espacio que posee, además le demuestra un porcentaje de ocupación en el disco virtual.

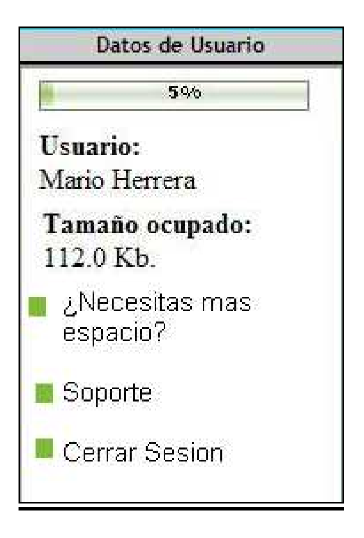
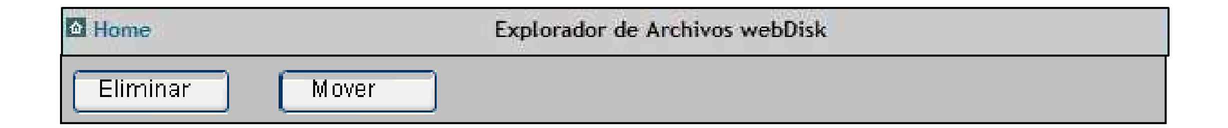


Figura # 9

10. Acción mover y eliminar

En la parte superior de la bandeja de entrada, en el Explorador de Archivos webdisk (Figura # 5) indica los botones de acción, como son:



10.1 Botón de acción Mover

Para mover un archivo o archivos consecutivos a cualquier carpeta, siga los siguientes pasos:

1. Selecciono el checkbox del archivo (paso #1)
2. Hago clic en el botón mover (paso # 2)
3. Me aparece el directorio de las carpeta existentes (paso #3)
4. Selecciono la carpeta donde se desplazará el archivo.

Automáticamente me aparece en el cuadro de texto el nombre de la carpeta y doy clic en aceptar.

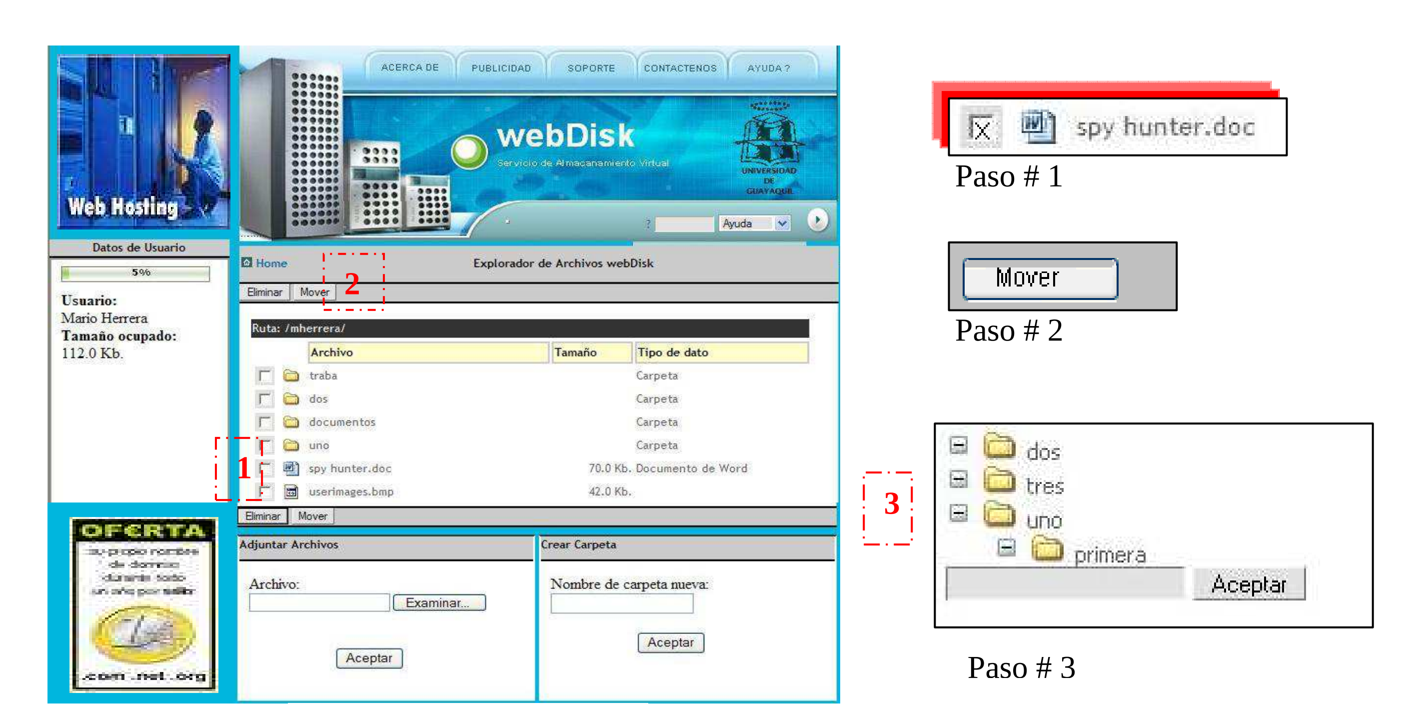
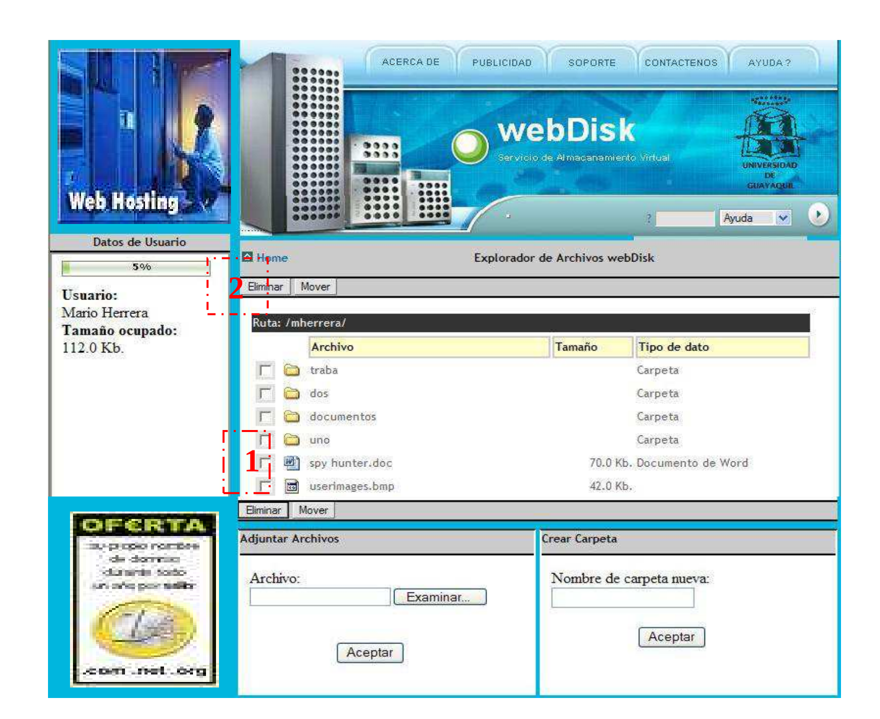


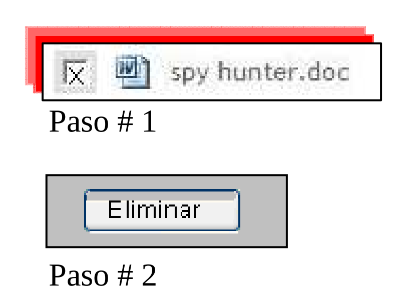
Figura # 10

10.2 Botón de acción Eliminar

Para eliminar un archivo, realizo los siguientes pasos:

1. Selecciono el checkbox del archivo (paso #1)
2. Hago clic en el botón eliminar (paso # 2).





3. Me aparece una ventana de alerta (figura # 11), indicando si deseo eliminar definitivamente el archivo o mantenerlo en la papelera de reciclaje.

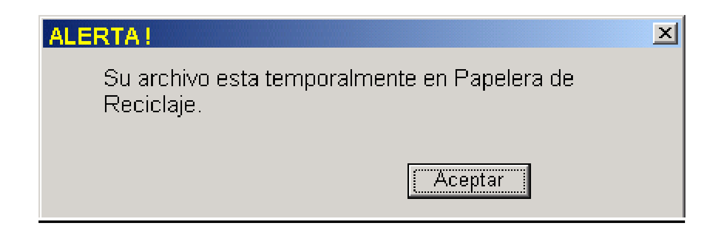


Figura # 11

11. Vinculos de la pagina web disk

11.1 Acerca del sitio web



Figura # 12

En este Link esta todo lo referente a Webdisk, como la licencia que es autorizada por la Universidad de Guayaquil, Copyright 2007 – 2008 y su primera versión 1.0

Además se podrá poner en contacto con los autores, indicando comentario o consultas, escriba en el cuadro de texto sus datos y su observación, y el administrador le orientará cualquier duda.

Califica este sitio Web, el usuario puede dar su aprobación por nuestro servicio, seleccionando el checkbox ya sea Excelente, Bueno, Regular o Malo. También su aprobación por la interfaz grafica que tal le pareció, seleccionando el checkbox en el casillero de Muy amigable, Fácil e intuitivo, Difícil.

Estas respuestas nos ayudaran a mejor el servicio o ver si el usuario esta satisfecho.

11.2 Publicidad

En este link estarán todas las empresas auspiciantes del servicio, ahí podrá encontrar las mejores ofertas de productos tecnológicos, mantenimiento y servicios, todo lo referente del mundo moderno.

11.3 Soporte

Dando un clic en el menú de soporte, encontrara el centro de seguridad para su equipo, como son las preguntas y respuestas:

- ✓ ¿Qué es un virus?
- ✓ ¿Cómo ingresa a la Pc?
- ✓ ¿Que es un Antivirus?
- ✓ ¿Que daños causan?
- ✓ Tipos de virus
- ✓ Medidas de seguridad

Dando un clic a cada pregunta encontrara automáticamente las mejores respuestas para usted.

11.4 Contáctenos

Finalizando con el detalle del menú soporte, encontraremos la opción Contáctenos, la misma que permitirá al usuario enviar sus comentarios y sugerencias vía mail.

En este menú se pondrá en contacto con los autores del proyecto, encontrara como comunicarse con ellos, como sus nombres, correo electrónico y teléfono. Ellos le ayudaran a resolver sus inquietudes y dudas referentes al servicio de la Webdisk.

11.5 Ayuda

11.5.1 Qué es Web Disk?

Es una herramienta que le permite almacenar archivos en la red y consultarlos desde cualquier lugar. Es similar a un disco rígido, con la ventaja que pode usarlo desde cualquier computadora, simplemente conectándote a Internet. Con el Web Disk podrá almacenar archivos, accederlos a ellos mediante una dirección Web desde cualquier lugar y compartirlos, a través de usuario y contraseña.

11.5.2. ¿Cómo funciona?

Web Disk le permite guardar todo tipo de archivos y luego consultarlos, mover y eliminar desde cualquier PC.

Además de disponer de 1.024 Mega Bytes para su uso, podrá crear carpetas y acceder a los archivos que guardo.

11.5.3. ¿Cuánto espacio tengo?

El espacio que tiene este gran disco rígido es de 1 Giga Byte.

11.5.4. ¿Cuánto cuesta?

Es totalmente gratis y el tiempo es ilimitado.

11.5.5. Preguntas Frecuentes

1. ¿Cómo se accede al Web Disk?
2. ¿Hay un límite de tamaño por archivo?
3. ¿Qué tipos de archivo puedo guardar en el Web Disk?
4. ¿Por cuánto tiempo es gratis, y si pago un valor puedo tener más espacio?

11.5.6.¿Cómo se accede al Web Disk?

Puede acceder a nuestro portal, desde la dirección: www.webdisk.udg.edu.ec

Una vez que ingresado al portal, le solicita el nombre de usuario y contraseña, tendrá que registrarse y luego podrá navegar en ella.

Luego aparece la bandeja de entrada con el nombre del usuario y podrá adjuntar todos los archivos que desee, solamente dando un clic en adjuntar archivos y para crear carpetas.

11.5.7. ¿Hay un límite de tamaño por archivo?

No. El límite está dado por el espacio disponible libre en el WebDisk.

11.5.8. ¿**Qué tipos de archivo puedo guardar en el Web Disk?**

Se puede guardar cualquier tipo de archivo. No hay ningún tipo de restricción.

11.5.9. ¿Por cuánto tiempo es gratis, y si pago un valor puedo tener más espacio?

Realmente es gratis y si desea mas espacio solicítelo al Administrador y el revisara si es uno de los usuarios concurrente que utiliza nuestro servicio, él le asignara mas espacio para que puede navegar y utilizar las funciones que ofrece la webdisk.udg.edu.ec